

# Formation Python - Algorithmique

**Objectifs :** Développer une expertise avancée en algorithmique avec Python en maîtrisant la complexité des algorithmes, les structures de données adaptées, la récursivité, les graphes et les arbres, afin d'optimiser les performances des programmes et de résoudre efficacement des problèmes complexes

**Compétences visées :** - Mesurer la complexité d'un algorithme avec la notation BigO

- Choisir les algorithmes et les structure de données les plus adaptés à ses problèmes et la volumétrie de ses données
- Appliquer des techniques d'optimisation de code et de réduction de complexité
- Utiliser la récursivité
- Manipuler efficacement des graphes et des arbres
- Rechercher et ordonner des données
- Appliquer des solutions à des problèmes complexes

**Durée :** 5 jour(s) (35 heures)

**Public :** Tout développeur ou scientifique ayant une expérience du langage Python

**Pré-requis :** Pour suivre ce stage dans de bonnes conditions, il est recommandé d'avoir suivi en amont la formation [Python - Bases et introduction aux bibliothèques scientifiques](#)

**Méthode pédagogique :** Pédagogie active mêlant exposés, exercices et applications pratiques dans le logiciel Python.

**Modalités d'évaluation :** Un formulaire d'auto-évaluation proposé en amont de la formation nous permettra d'évaluer votre niveau et de recueillir vos attentes. Ce même formulaire soumis en aval de la formation fournira une appréciation de votre progression.

Des exercices pratiques seront proposés à la fin de chaque séquence pédagogique pour l'évaluation des acquis.

En fin de formation, vous serez amené(e) à renseigner un questionnaire d'évaluation à chaud.

Une attestation de formation vous sera adressée à l'issue de la session.

Trois mois après votre formation, vous recevrez par email un formulaire d'évaluation à froid sur l'utilisation des acquis de la formation.

**Accessibilité :** Vous souhaitez suivre notre formation Formation par ville et êtes en situation de handicap ? Merci de nous contacter afin que nous puissions envisager les adaptations nécessaires et vous garantir de bonnes conditions d'apprentissage

**Tarifs :**

- Présentiel : 3250 € HT
  - Distanciel : 3000 € HT
- (-10% pour 2 inscrits, -20% dès 3 inscrits)

**Option(s) :**

- Forfait déjeuners : 100 € HT

## Nos prochaines sessions

### Distance

du 2 au 6 juin 2025

du 24 au 28 novembre 2025

### Lyon

du 19 au 23 mai 2025

du 15 au 19 décembre 2025

### Paris

du 23 au 27 juin 2025

du 8 au 12 décembre 2025

### Toulouse

du 6 au 10 octobre 2025

## Programme :

### - Complexité des algorithmes : 0,5 jour

- Comprendre la notation BigO
- Mesurer la complexité d'un algorithme
- Premières techniques pour diminuer la complexité d'un algorithme

*Travaux pratiques :*

*Calcul et vérification par la mesure de la complexité de plusieurs algorithmes. Simplification d'un algorithme de complexité quadratique pour le rendre linéaire.*

### - Rappels et manipulation de base des structures de données: 0,5 jour

- Tableaux indicés
- Tableaux associatifs/dictionnaires
- Files et piles
- Skip list
- Listes chaînées
- Arbres et graphes

*Travaux pratiques :*

*Création, modification, recherche, insertion, suppression d'éléments avec chaque structure et mesure de leurs complexités respectives.*

## - Récursivité: 0,5 jour

- Prise en mains et cas d'utilisations: suites numériques, analyseurs syntaxiques, parcours d'arbres, ...
- Bien comprendre la récursivité
- Mise en oeuvre top down et bottom up
- Limites

*Travaux pratiques :*

*Implémentation et explication de nombreux cas pratiques: suite numérique, tour de Hanoï, création d'un analyseur syntaxique, trouver toutes les solutions du jeu le compte est bon*

## - Les graphes: 1,5 jour

- Définition d'un graphe, nœuds, arrêtes, directionnels, ...
- Pourquoi utiliser un graphe
- Exemples d'algorithmes : recherche de relations, du plus court/long chemin, problème du voyageur de commerce, ...
- Matrice d'adjacence
- Visualiser un graphe
- Parcourir un graphe
- Recherche dans un graphe
- Graphes pondérés
- Présentation et mise en œuvre d'algorithmes: Dijkstra, chemins hamiltoniens, 2-opt, ...

*Travaux pratiques:*

*Implémentation de la recherche du plus court chemin (et son tracé) dans un graphe de dimension réelle de toutes les rues de Paris. Recherche et parcours d'informations.*

## - Les arbres: 1 jour

- Définition d'un arbre
- Présentation des arbres binaires: parcours en ordre, préfixé et postfixé
- Recherche dans un arbre binaire
- Insertion et suppression
- Utilisation d'arbres Trie/m-aire

*Travaux pratiques :*

*Création, manipulation et recherche dans des arbres.*

## - Application à des problèmes réels : 1 jour

- Mise en œuvre de différents algorithmes de tri et de recherche
- Création d'un analyseur syntaxique et d'un interpréteur
- Résolution d'anagrammes
- Jeu du labyrinthe : génération et résolution

*Travaux pratiques :*

*De nombreux autres exercices et jeux : astéroïdes, somme maximale dans une pyramide, recherche d'un point d'équilibre, programmation sans branche, somme de pièces, mémorisation, la mine d'or,...*

*Date de dernière modification : 17 mars 2025*